

QQ normality plots

Harvey Motulsky, GraphPad Software Inc.
July 2013

Introduction

Many statistical tests assume that data (or residuals) are sampled from a Gaussian distribution. Normality tests are often used to test for evidence that the distribution is not Gaussian, but these tests are less helpful than many scientists expect them to be.

Many books (but not [mine](#)) and programs (but not [GraphPad Prism yet](#)) suggest using a QQ normality plot to visually assess whether or not a distribution appears to be sampled from a Gaussian distribution. The problem is that most books give very complicated explanations of QQ plots, making it very hard to figure out what exactly they mean. In fact, the basic idea of a QQ normality plot is pretty easy to understand.

What is a QQ normality plot?

Figure 1 is an example of a QQ normality plot

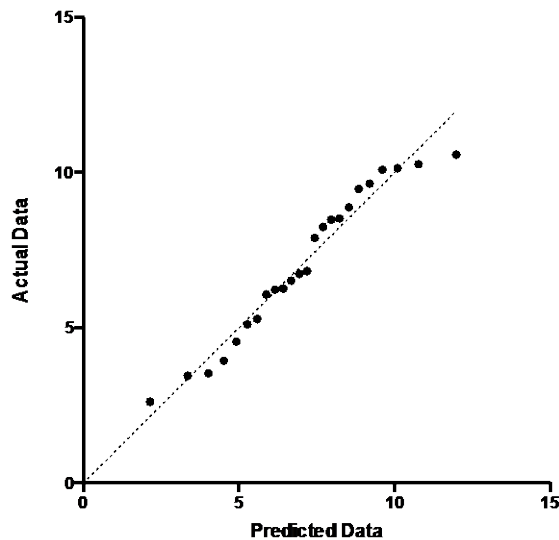


Figure 1

The Y axis plots the actual values. The X axis plots predicted values. To calculate these predicted values, first assume the distribution is Gaussian (with the mean and SD of the data), and then select n points (the number in the data set) from that Gaussian distribution with equally spaced percentiles. For example, if there are five values in the sample choose values from the Gaussian distribution at the 10th, 30th,

50th, 70th and 90th percentiles. Each predicted value becomes a Y value matched by rank with one of the actual values which is the corresponding X value. This algorithm is explained in more detail below.

How to interpret a QQ normality plot

If the data are truly sampled from a Gaussian distribution, the QQ plot will be linear. If the X and Y values are comparable (as they are in Figure 1), then the points are expected to line up on the line of identity (shown in a dotted line in Figure 1).

Systematic deviation from this ideal is evidence that the data are not sampled from a Gaussian distribution. But it is hard to say how much deviation is more than you'd expect to see by chance. It takes some experience to interpret a QQ plot.

The data in Figure 1 were sampled (by computer) from a Gaussian distribution with a mean equal to 7.0 and a SD equal to 2.0. I repeated this simulation 20 times (with different random values). Most of those simulations created graphs very similar to Figure 1. Figure 2 below shows the four (of twenty) graphs that seem to deviate the most from linearity.

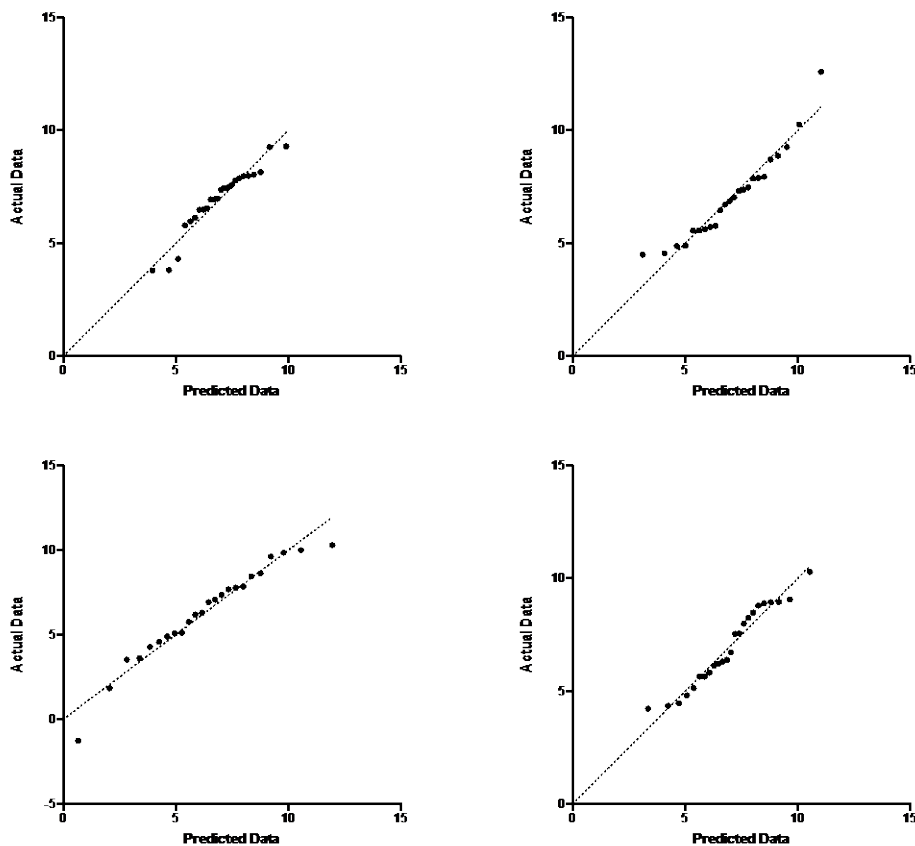


Figure 2

What does a QQ plot look like when the data are sampled from a lognormal distribution?

[Lognormal distributions](#) are common in biology. To simulate a lognormal distribution, I sampled data from a Gaussian distribution with a mean equal to 2.0 and a SD equal to 0.5 (arbitrary choices). I then transformed these data to their antilogarithm (10^Y), which creates a sample of data sampled from a lognormal distribution. I then created a QQ plot (Figure 3). The data systematically and consistently deviate from the line of identity, so you can instantly see that these data were not sampled from a Gaussian distribution.

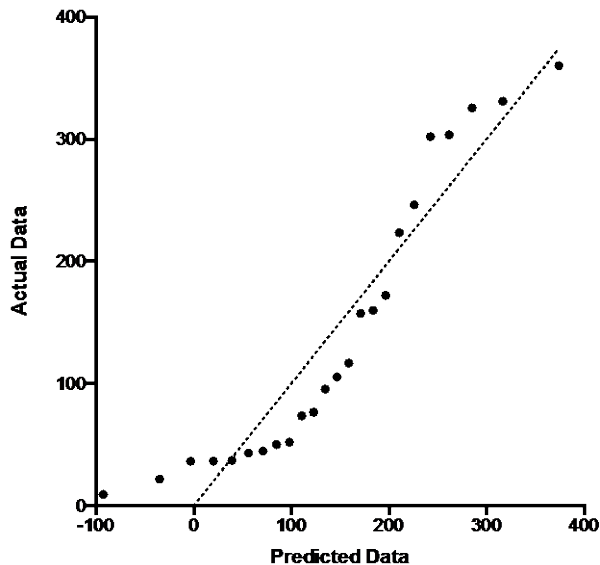


Figure 3

Math details

If you are curious about the details, read on. Otherwise, skip this section. To calculate the predicted values, follow these steps:

1. Compute a *percentile* for each value in the data set. One (of several) methods to compute a percentile from a value is to compute $100(i-0.5)/n$, where i is the rank (1 is the lowest) and n is the sample size. Figure 1 has 24 values, so the percentile for the fifth smallest value is $100*4.5/24= 18.75$.
2. For each percentile, calculate how many standard deviations from the mean you need to go to reach that percentile on a Gaussian distribution. Use the built-in in `NORMSINV()` function in Excel, or [this user-defined Prism function](#). We call the result the *rank based z score*. The median has a rank based z score of 0, values larger than the median have positive z scores, and values smaller than the median have negative z scores. For the 18.78 percentile, the ranked based z score is -0.886 (computed by the excel formula `=normsinv(0.1878)`).

- This means that 18.78 percent of a standard normal distribution (mean=0, SD=1.0) is less than -0.886.
3. Compute corresponding values from a normal distribution whose mean and SD come from the actual data. Multiply each rank-based z score, computed in step 2, times the actual SD of the data. Then add that value, which might be positive or negative, to the actual mean of the data. The values in Figure 1 have a mean equal to 7.057 and a SD equal to 2.415. So the predicted value for the fifth smallest value is $-0.886 * 2.415 + 7.057 = 4.92$.
 4. Match each predicted value with an actual value by rank and graph. The fifth smallest value in the data set is 4.56. So one of the points on the QQ normal plot is $X=4.56, Y=4.92$.

Q&A

Q. What does QQ stand for?

A. Quantile-quantile.

Q. What is a quantile?

A. It is the same as a percentile, but expressed as a fraction. The 95th percentile is equivalent to a quantile of 0.95. The median, or 50th percentile, is 0.50 expressed as a quantile. In the explanation above, I used percentiles but most books use quantiles.

Q. So why doesn't the graph show quantiles or percentiles?

A. The name is an historical carryover, as the plots one usually sees now don't show quantiles on either axis. But quantiles (or percentiles as explained above) are used in the calculations that generate the predicted values.

Q. Why do different programs give slightly different results?

A. Three reasons:

- There are many ways to compute a quantile from a set of values. Wikipedia [lists many other variations](#).
- Different programs may handle ties (two or more identical values) differently.
- Different programs may use different approximations to the inverse normal distribution.

Q. Other books seem to describe computing a different value step 2 above?

A. The terminology is confusing. The value computed in step 2 goes by many names, including:

- Rank based z scores
- Inverse normal
- Unit normal quantile
- Normal quantiles
- Theoretical quantiles
- Normal theoretical quantiles
- Rankit

Q. Is a normal distribution the same as a Gaussian distribution?

A. Yes.

Q. Why do different programs plot different graphs under the name normal QQ plot?

A. There seems to be no strong standard for which set of values goes on which axis. The graphs are sometimes made with X and Y opposite of Figure 1, with the observed data on the X axis and the computed values (the predictions) plotted on the Y axis.

The values plotted of the two axes can vary as well. There are three variations:

- Actual data vs. corresponding values computed from the Gaussian distribution (Figure 1). This is the easiest to understand.
- Actual data vs. computed rank based z scores (explained at the end of the article)
- Actual z scores vs. rank based z scores.

If you only look at the pattern of the data points, all three of graphs will look the same. Since the X and Y values may be computed differently than Figure 1, the scale of the values (numbers labeling the axes) will be different. In the second bullet point above, the X and Y axis are in different units, so it makes no sense to plot the line of identity.

Creating a QQ plot with Prism

Prism (up to and including version 6) was not designed to create QQ plots, but can do so with some effort. Figure 1 was made with Prism using [this file](#). Inspecting this file will also show you use of three useful Prism features you may not know about:

- Hooked analysis constants. The column statistics analysis computes the mean, SD and sample size (n). These values are then hooked into analyses. The analysis that computes the rank based z score uses n. The analysis that converts those scores to the data units uses the mean and SD hooked from column statistics.
- Paste linking. The table used to make the graph contain two columns of values copied from two different tables. The copied values were paste linked, which means they update when the original data are edited or replaced.
- User-defined transforms. The file uses one to find values on a Gaussian distribution using the [Normsinv function](#).

Figure 2 was made with [this file](#), and Figure 3 with [this one](#).